

# On the Simulatability Condition in Key Generation Over a Non-authenticated Public Channel

Wenwen Tu and Lifeng Lai

## Abstract

Simulatability condition is a fundamental concept in studying key generation over a non-authenticated public channel, in which Eve is active and can intercept, modify and falsify messages exchanged over the non-authenticated public channel. Using this condition, Maurer and Wolf showed a remarkable “all or nothing” result: if the simulatability condition does not hold, the key capacity over the non-authenticated public channel will be the same as that of the case with a passive Eve, while the key capacity over the non-authenticated channel will be zero if the simulatability condition holds. However, two questions remain open so far: 1) For a given joint probability mass function (PMF), are there efficient algorithms (polynomial complexity algorithms) for checking whether the simulatability condition holds or not?; and 2) If the simulatability condition holds, are there efficient algorithms for finding the corresponding attack strategy? In this paper, we answer these two open questions affirmatively. In particular, for a given joint PMF, we construct a linear programming (LP) problem and show that the simulatability condition holds *if and only if* the optimal value obtained from the constructed LP is zero. Furthermore, we construct another LP and show that the minimizer of the newly constructed LP is a valid attack strategy. Both LPs can be solved with a polynomial complexity.

## Index Terms

Active adversary, Computational complexity, Farkas’ lemma, Linear programming, Non-authenticated channel, Simulatability condition.

W. Tu and L. Lai are with Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA. Email: {wtu, llai}@wpi.edu. The work of W. Tu and L. Lai was supported by the National Science Foundation CAREER Award under Grant CCF-1318980 and by the National Science Foundation under Grant CNS-1321223.

## I. INTRODUCTION

The problem of secret key generation via public discussion under both source and channel models has attracted significant research interests [1]–[11]. Under the source model, users observe correlated sources generated from a certain joint probability mass function (PMF), and can discuss with each other via a noiseless public channel. Any discussion over the public channel will be overheard by Eve. Furthermore, the public channel can either be authenticated or non-authenticated. An authenticated public channel implies that Eve is a passive listener. On the other hand, a non-authenticated public channel implies that Eve is active and can intercept, modify or falsify any message exchanged through the public channel.

Clearly, the secret key rate that can be generated using the non-authenticated public channel is no larger than that can be generated using the authenticated public channel. In [8]–[11], Maurer and Wolf introduced a concept of simulatability condition (this condition will be defined precisely in the sequel) and established a remarkable “all or nothing” result. In particular, they showed that for the secret key generation via a non-authenticated public channel with two legitimate terminals in the presence of an active adversary: 1) if the simulatability condition holds, the two legitimate terminals will not be able to establish a secret key, and hence the key capacity is 0; and 2) if the simulatability condition does not hold, the two legitimate terminals can establish a secret key and furthermore the key capacity will be the same as that of the case when Eve is passive. Intuitively speaking, if the simulatability condition holds, from its own source observations, Eve can generate fake messages that are indistinguishable from messages generated from legitimate users. On the other hand, if the simulatability condition does not hold, the legitimate users will be able to detect modifications made by Eve.

It is clear that the simulatability condition is a fundamental concept for the key generation via a non-authenticated public channel, and hence it is important to design efficient algorithms to check whether the simulatability condition holds or not. Using ideas from mechanical models, [10] made significant progress in designing efficient algorithms. In particular, [10] proposed to represent PMFs as mass constellations in a coordinate, and showed that the simulatability condition holds if and only if one mass constellation can be transformed into another mass constellation using a finite number of basic mass operations. Furthermore, [10] introduced another notion of one mass constellation being “more centered” than another constellation and designed a

low-complexity algorithm to check this “more centered” condition. For some important special cases, which will be described precisely in Section II, [10] showed that the “more centered” condition is necessary and sufficient for the mass constellation transformation problem (and hence is necessary and sufficient condition for the simulatability condition for these special cases). However, in the general case, the “more centered” condition is a necessary but not sufficient condition for the mass constellation transformation problem. Hence, whether there exists efficient algorithms for the mass constellation transformation problem (and hence the simulatability condition) in the general case is still an open question.

As the result, despite the significant progress made in [10], the following two questions remain open regarding the simulatability condition for the general case:

- 1) For a given joint PMF, are there efficient algorithms (polynomial complexity algorithms) for checking whether the simulatability condition holds or not?
- 2) If the simulatability condition holds, are there efficient algorithms for finding the corresponding Eve’s attack strategy?

In this paper, we answer these two open questions affirmatively.

To answer the first open question, we construct a linear programming (LP) problem and show that the simulatability condition holds **if and only if** the optimal value obtained from this LP is zero. We establish our result in three main steps. We first show that, after some basic transformations, checking whether the simulatability condition holds or not is equivalent to checking whether there exists a nonnegative solution to a specially constructed system of linear equations. We then use a basic result from linear algebra to show that whether there exists a nonnegative solution to the constructed system of linear equations is equivalent to whether there is a solution (not necessarily nonnegative) to a related system of inequalities or not. Finally, we use Farkas’ lemma [12], a fundamental result in linear programming and other optimization problems, to show that whether the system of inequalities has a solution or not is equivalent to whether the optimal value of a specially constructed LP is zero or not. Since there exists polynomial complexity algorithms for solving LP problems [13]–[15], we thus find a polynomial complexity algorithm for checking the simulatability condition for a general PMF.

To answer the second open question, we construct another LP and show that the minimizer of this LP is a valid attack strategy. The proposed approach is very flexible in the sense that one can simply modify the cost function of the constructed LP to obtain different attack strategies.

Furthermore, the cost function can be modified to satisfy various design criteria. For example, a simple cost function can be constructed to minimize the amount of modifications Eve needs to perform during the attack. All these optimization problems with different cost functions can be solved with a polynomial complexity.

The remainder of the paper is organized as follows. In Section II, we introduce some preliminaries and the problem setup. In Section III, we present our main results. In Section IV, we use numerical examples to illustrate the proposed algorithm. In Section V, we present an approach to further reduce the computational complexity. In Section VI, we offer our concluding remarks.

## II. PRELIMINARIES AND PROBLEM SETUP

Let  $\mathcal{X} = \{1, \dots, |\mathcal{X}|\}$ ,  $\mathcal{Y} = \{1, \dots, |\mathcal{Y}|\}$  and  $\mathcal{Z} = \{1, \dots, |\mathcal{Z}|\}$  be three finite sets. Consider three correlated random variables  $(X, Y, Z)$ , taking values from  $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ , with joint PMF  $P_{XYZ}$ , the simulatability condition is defined as follows:

**Definition 1.** ([8]) *For a given  $P_{XYZ}$ , we say  $X$  is simulatable by  $Z$  with respect to  $Y$ , denoted by  $\text{Sim}_Y(Z \rightarrow X)$ , if there exists a conditional PMF  $P_{\bar{X}|Z}$  such that  $P_{Y\bar{X}} = P_{YX}$ , with*

$$P_{Y\bar{X}}(y, x) = \sum_{z \in \mathcal{Z}} P_{YZ}(y, z) \cdot P_{\bar{X}|Z}(x|z), \quad (1)$$

in which  $P_{YX}$  and  $P_{YZ}$  are the joint PMFs of  $(Y, X)$  and  $(Y, Z)$  under  $P_{XYZ}$  respectively.

One can also define  $\text{Sim}_X(Z \rightarrow Y)$  in the same manner. This concept of simulatability, first defined in [8], is a fundamental concept in the problem of secret key generation over a non-authenticated public channel [9]–[11], in which two terminals Alice and Bob would like to establish a secret key in the presence of an adversary Eve. These three terminals observe sequences  $X^N$ ,  $Y^N$  and  $Z^N$  generated according to

$$P_{X^N Y^N Z^N}(x^N, y^N, z^N) = \prod_{i=1}^N P_{XYZ}(x_i, y_i, z_i). \quad (2)$$

Alice and Bob can discuss with each other via a public **non-authenticated** noiseless channel, which means that Eve not only has full access to the channel but can also interrupt, modify and falsify messages exchanged over this public channel. The largest key rate that Alice and Bob<sup>1</sup> can generate with the presence of the active attacker is denoted as  $S^*(X; Y||Z)$ . Let

<sup>1</sup>Please see [9]–[11] for precise definitions.

$S(X; Y||Z)$  denote the largest key rate that Alice and Bob can generate when Eve is passive, i.e., when the public channel is authenticated. Clearly,  $S(X; Y||Z) \geq S^*(X; Y||Z)$ . Although a full characterization of  $S(X; Y||Z)$  is unknown in general, [9] established the following remarkable “all or nothing” result:

**Theorem 1.** ([9]) *If  $\text{Sim}_Y(Z \rightarrow X)$  or  $\text{Sim}_X(Z \rightarrow Y)$ , then  $S^*(X; Y||Z) = 0$ . Otherwise,  $S^*(X; Y||Z) = S(X; Y||Z)$ .*

This significant result implies that, if the simulatability condition does not hold, one can generate a key with the same rate as if Eve were passive. On the other hand, if the simulatability condition holds, the key rate will be zero. Intuitively speaking, if  $\text{Sim}_Y(Z \rightarrow X)$  holds, then after observing  $Z^N$ , Eve can generate  $\bar{X}^N$  by passing  $Z^N$  through a channel defined by  $P_{\bar{X}|Z}$ . Then  $(\bar{X}^N, Y^N)$  has the same statistics as  $(X^N, Y^N)$ . Hence by knowing only  $Y^N$ , Bob cannot distinguish  $\bar{X}^N$  and  $X^N$ , and hence cannot distinguish Alice or Eve.

As mentioned in the introduction, [10] has made important progress in developing low-complexity algorithms for checking whether  $\text{Sim}_Y(Z \rightarrow X)$  (or  $\text{Sim}_X(Z \rightarrow Y)$ ) holds or not. In particular, [10] developed an efficient algorithm to check a related condition called “more centered” condition. When  $|\mathcal{Y}| = 2$ , that is when  $Y$  is a binary random variable, this “more centered” condition is shown to be necessary and sufficient for  $\text{Sim}_Y(Z \rightarrow X)$ . Hence, [10] has found an efficient algorithm to check  $\text{Sim}_Y(Z \rightarrow X)$  for the special case of  $Y$  being binary (the algorithm is also effective in checking  $\text{Sim}_X(Z \rightarrow Y)$  when  $X$  is binary). However, when  $Y$  is not binary, the “more centered” condition is only a necessary condition for  $\text{Sim}_Y(Z \rightarrow X)$ . Hence, two questions remain open:

- 1) For a general given  $P_{XYZ}$ , are there efficient algorithms (polynomial complexity algorithms) for checking whether  $\text{Sim}_Y(Z \rightarrow X)$  (or  $\text{Sim}_X(Z \rightarrow Y)$ ) holds or not?
- 2) If  $\text{Sim}_Y(Z \rightarrow X)$  (or  $\text{Sim}_X(Z \rightarrow Y)$ ) holds, are there efficient algorithms for finding the corresponding  $P_{\bar{X}|Z}$  (or  $P_{\bar{Y}|Z}$ )?

In this paper, we answer these two open questions affirmatively.

*Notations:* Throughout this paper, we use boldface uppercase letters to denote matrices, boldface lowercase letters to denote vectors. We also use  $\mathbf{1}$ ,  $\mathbf{0}$  and  $\mathbf{I}$ , unless stated otherwise, to denote all ones column vector, all zeros column vector and the identity matrix, respectively. In addition, we denote the vectorization of a matrix by  $\text{Vec}(\cdot)$ . Specifically, for an  $m \times n$  matrix

$\mathbf{A}$ ,  $\text{Vec}(\mathbf{A})$  is an  $mn \times 1$  column vector:

$$\text{Vec}(\mathbf{A}) = [a_{11}, \dots, a_{m1}, \dots, a_{1n}, \dots, a_{mn}]^T, \quad (3)$$

in which  $[\cdot]^T$  is the transpose of the matrix. And vice versa can be done by  $\mathbf{A} = \text{Reshape}(\text{Vec}(\mathbf{A}), [m, n])$ .

We use  $\mathbf{A} \otimes \mathbf{B}$  to denote the Kronecker product of matrices  $\mathbf{A}$  and  $\mathbf{B}$ . Specifically, assume  $\mathbf{A}$  is an  $m \times n$  matrix, then

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}. \quad (4)$$

All matrices and vectors in this paper are real.

### III. MAIN RESULTS

In this paper, we focus on  $\text{Sim}_Y(Z \rightarrow X)$ . The developed algorithm can be easily modified to check  $\text{Sim}_X(Z \rightarrow Y)$ . We rewrite (1) in the following matrix form

$$\mathbf{C} = \mathbf{A}\mathbf{Q}, \quad (5)$$

in which  $\mathbf{C} = [c_{ij}]$  is a  $|\mathcal{Y}| \times |\mathcal{X}|$  matrix with  $c_{ij} = P_{YX}(i, j)$ ,  $\mathbf{A} = [a_{ik}]$  is a  $|\mathcal{Y}| \times |\mathcal{Z}|$  matrix with  $a_{ik} = P_{YZ}(i, k)$ , and  $\mathbf{Q} = [q_{kj}]$  is a  $|\mathcal{Z}| \times |\mathcal{X}|$  matrix with  $q_{kj} = P_{\tilde{X}|Z}(j|k)$  if such  $P_{\tilde{X}|Z}$  exists.

Checking whether  $\text{Sim}_Y(Z \rightarrow X)$  holds or not is equivalent to checking whether there exists a transition matrix  $\mathbf{Q}$  such that (5) holds. As  $\mathbf{Q}$  is a transition matrix, its entries  $q_{kj}$ s must satisfy

$$q_{kj} \geq 0, \quad \forall k \in [1 : |\mathcal{Z}|], j \in [1 : |\mathcal{X}|], \quad (6)$$

$$\sum_{j=1}^{|\mathcal{X}|} q_{kj} = 1, \quad \forall k \in [1 : |\mathcal{Z}|]. \quad (7)$$

We note that if  $q_{kj}$ s satisfy (6) and (7), they will automatically satisfy  $q_{kj} \leq 1$ . Hence, we don't need to state this requirement here.

If there exists at least one transition matrix  $\mathbf{Q}$  satisfying (5), (6) and (7) simultaneously, we can conclude that the simulatability condition  $\text{Sim}_Y(Z \rightarrow X)$  holds.

(7) can be written in the matrix form

$$\mathbf{1}_{|\mathcal{Z}| \times 1} = \mathbf{Q}\mathbf{1}_{|\mathcal{X}| \times 1}, \quad (8)$$

Then, (5) and (8) can be written in the following compact form:

$$\begin{aligned}
& \begin{bmatrix} \text{Vec}(\mathbf{C}^T) \\ \mathbf{1}_{|Z| \times 1} \end{bmatrix} \\
&= \begin{bmatrix} a_{11}\mathbf{I} & a_{12}\mathbf{I} & \cdots & a_{1|Z|}\mathbf{I} \\ \vdots & \vdots & \ddots & \vdots \\ a_{|\mathcal{Y}|1}\mathbf{I} & a_{|\mathcal{Y}|2}\mathbf{I} & \cdots & a_{|\mathcal{Y}||Z|}\mathbf{I} \\ \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \end{bmatrix} \text{Vec}(\mathbf{Q}^T) \\
&= \begin{bmatrix} \mathbf{A} \otimes \mathbf{I} \\ \mathbf{I}_{|Z|} \otimes \mathbf{1} \end{bmatrix} \text{Vec}(\mathbf{Q}^T), \tag{9}
\end{aligned}$$

in which the sizes for  $\mathbf{I}$ ,  $\mathbf{1}$  and  $\mathbf{0}$  are  $|\mathcal{X}| \times |\mathcal{X}|$ ,  $1 \times |\mathcal{X}|$  and  $1 \times |\mathcal{X}|$ , respectively.

For notational convenience, we define

$$\mathbf{c} \triangleq \begin{bmatrix} \text{Vec}(\mathbf{C}^T) \\ \mathbf{1}_{|Z| \times 1} \end{bmatrix}, \tag{10}$$

$$\mathbb{A} \triangleq \begin{bmatrix} \mathbf{A} \otimes \mathbf{I} \\ \mathbf{I}_{|Z|} \otimes \mathbf{1} \end{bmatrix}, \tag{11}$$

$$\mathbf{q} \triangleq \text{Vec}(\mathbf{Q}^T). \tag{12}$$

From (9), it is clear that  $\mathbf{c}$  is an  $m \times 1$  vector,  $\mathbb{A}$  is an  $m \times n$  matrix, and  $\mathbf{q}$  is an  $n \times 1$  vector, in which

$$m = |\mathcal{Y}||\mathcal{X}| + |Z|, \tag{13}$$

$$n = |Z||\mathcal{X}|. \tag{14}$$

With these notation and combining (9) with (6), the original problem of checking whether  $\text{Sim}_Y(Z \rightarrow X)$  holds or not is equivalent to checking whether there exists **nonnegative** solutions  $\mathbf{q}$  for the system

$$\mathbb{A}\mathbf{q} = \mathbf{c}. \tag{15}$$

In the following, we check whether there exists at least a nonnegative solution for the system defined by (15). There are two main steps: 1) whether the system is consistent or not; 2) if it is consistent, whether there exists a nonnegative solution or not. Checking the consistency of (15) is straightforward: a necessary and sufficient condition for a system of non-homogenous linear equations to be consistent is

$$\text{Rank}(\mathbb{A}) = \text{Rank}((\mathbb{A}|\mathbf{c})), \quad (16)$$

where  $(\mathbb{A}|\mathbf{c})$  is the augmented matrix of  $\mathbb{A}$ . If (16) is not satisfied, it can be concluded that  $\text{Sim}_Y(Z \rightarrow X)$  does not hold. If (16) is satisfied, we need to further check whether there exists a nonnegative solution to (15) or not.

To proceed further, we will need the following definition of generalized inverse (g-inverse) of a matrix  $\mathbf{G}$ .

**Definition 2.** ([16]) *For a given  $m \times n$  real matrix  $\mathbf{G}$ , an  $n \times m$  real matrix  $\mathbf{G}^g$  is called a g-inverse of  $\mathbf{G}$  if*

$$\mathbf{G}\mathbf{G}^g\mathbf{G} = \mathbf{G}.$$

The g-inverse  $\mathbf{G}^g$  is generally not unique (If  $n = m$  and  $\mathbf{G}$  is full rank, then  $\mathbf{G}^g$  is unique and equal to the inverse matrix  $\mathbf{G}^{-1}$ ). A particular choice of g-inverse is called the Moore-Penrose pseudoinverse  $\mathbf{G}^+$ , which can be computed using multiple different approaches. One approach is to use the singular value decomposition (SVD): by SVD, for a given  $\mathbf{G}$  and its SVD decomposition

$$\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (17)$$

then,  $\mathbf{G}^+$  can be obtained as

$$\mathbf{G}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T, \quad (18)$$

in which  $\mathbf{\Sigma}^+$  is obtained by taking the reciprocal of each non-zero element on the diagonal of the diagonal matrix  $\mathbf{\Sigma}$ , leaving the zeros in place. One can easily check that the Moore-Penrose pseudoinverse  $\mathbf{G}^+$  obtained by SVD satisfies the g-inverse matrix definition and hence is a valid g-inverse.

With the concept of g-inverse, we are ready to state our main result regarding the first open question.



**Theorem 2.** Let  $\mathbb{A}^g$  be any given  $g$ -inverse of  $\mathbb{A}$  (e.g., it can be chosen as the Moore-Penrose pseudoinverse  $\mathbb{A}^+$ ), and  $h^*$  be obtained by the following LP

$$\begin{aligned} h^* &= \min_{\mathbf{t}} \{\mathbf{t}^T \mathbb{A}^g \mathbf{c}\}, \\ \text{s. t.} \quad \mathbf{t} &\succeq \mathbf{0}, \\ (\mathbf{I} - \mathbb{A}^g \mathbb{A})^T \mathbf{t} &= \mathbf{0}. \end{aligned} \tag{19}$$

Then  $\text{Sim}_Y(Z \rightarrow X)$  holds, **if and only if**  $h^* = 0$  and (16) holds.

*Proof:* If (16) does not hold, then there is no solution to (15), and hence  $\text{Sim}_Y(Z \rightarrow X)$  does not hold.

In the remainder of the proof, we assume that (16) holds. If (16) holds, the general solution to (15) can be written in the following form (see, e.g., Theorem 2 a.(d) of [17])

$$\mathbf{q} = \mathbb{A}^g \mathbf{c} + (\mathbb{A}^g \mathbb{A} - \mathbf{I}) \mathbf{p}, \tag{20}$$

in which  $\mathbb{A}^g$  can be any given  $g$ -inverse of  $\mathbb{A}$ , and  $\mathbf{p}$  is an arbitrary length- $n$  vector.

As the result, the problem of whether there exists a nonnegative solution to (15) (i.e.,  $\mathbf{q} \succeq \mathbf{0}$ ) is equivalent to the problem of whether there exists a solution  $\mathbf{p}$  for the following system defined by

$$(\mathbf{I} - \mathbb{A}^g \mathbb{A}) \mathbf{p} \preceq \mathbb{A}^g \mathbf{c}. \tag{21}$$

To check whether the system defined by (21) has a solution, we use Farkas' lemma, a fundamental lemma in linear programming and related area in optimization. For completeness, we state the form of Farkas' lemma used in our proof in Appendix A. To use Farkas' lemma, we first write a LP related to the system defined in (21)

$$\begin{aligned} h^* &= \min_{\mathbf{t}} \{\mathbf{t}^T \mathbb{A}^g \mathbf{c}\}, \\ \text{s.t.} \quad \mathbf{t} &\succeq \mathbf{0}, \\ (\mathbf{I} - \mathbb{A}^g \mathbb{A})^T \mathbf{t} &= \mathbf{0}. \end{aligned}$$

The above LP is always feasible since  $\mathbf{t} = \mathbf{0}$  is a vector that satisfies the constraints, which results in  $\mathbf{t}^T \mathbb{A}^g \mathbf{c} = 0$ . Hence the optimal value  $h^* \leq 0$ . Using Farkas' lemma, we have that (21) has a solution **if and if**  $h^* = 0$ . More specifically, if  $h^* = 0$ , then there exists at least a

solution  $\mathbf{p}$  for (21), which further implies that there is a nonnegative solution to (15), and hence  $\text{Sim}_Y(Z \rightarrow X)$  holds. On the other hand, if  $h^* < 0$ , then there is no solution  $\mathbf{p}$  for (21), which further implies that there is no nonnegative solution to (15), and hence  $\text{Sim}_Y(Z \rightarrow X)$  does not hold.  $\blacksquare$

As mentioned above, if  $\text{Rank}(\mathbb{A}) = m = n$  holds, then  $\mathbb{A}^g = \mathbb{A}^{-1}$  is unique. For other cases,  $\mathbb{A}^g$  might not be unique. One may wonder whether different choices of  $\mathbb{A}^g$  will affect the result in Theorem 2 or not. The following proposition answers this question.

**Proposition 1.** *Different choices of  $\mathbb{A}^g$  will not affect the result on whether  $h^*$  equals 0 or not.*

*Proof:* Let  $\mathbb{A}_1^g$  and  $\mathbb{A}_2^g$  be two different  $g$ -inverses of  $\mathbb{A}$ , and let  $h_1^*$  and  $h_2^*$  be the values obtained using  $\mathbb{A}_1^g$  and  $\mathbb{A}_2^g$  in (19) respectively. It suffices to show that if  $h_1^* = 0$ , then  $h_2^* = 0$ .

Assuming that  $h_1^* = 0$ , then there exists a vector  $\mathbf{p}_1$  satisfying  $(\mathbf{I} - \mathbb{A}_1^g \mathbb{A})\mathbf{p}_1 \preceq \mathbb{A}_1^g \mathbf{c}$ , we will show that there exists a vector  $\mathbf{p}_2$  satisfying  $(\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_2 \preceq \mathbb{A}_2^g \mathbf{c}$ , which then implies  $h_2^* = 0$ .

First, we know that  $\mathbb{A}_1^g \mathbf{c}$  and  $\mathbb{A}_2^g \mathbf{c}$  are two solutions to the system  $\mathbb{A}\mathbf{q} = \mathbf{c}$ , which can be easily verified by setting  $\mathbb{A}^g$  as  $\mathbb{A}_1^g$  and  $\mathbb{A}_2^g$  in (20) respectively and setting  $\mathbf{p} = \mathbf{0}$ . This implies that

$$\mathbb{A}(\mathbb{A}_2^g \mathbf{c} - \mathbb{A}_1^g \mathbf{c}) = \mathbf{0}, \quad (22)$$

and hence  $\mathbb{A}_2^g \mathbf{c} - \mathbb{A}_1^g \mathbf{c}$  is a solution to the system  $\mathbb{A}\mathbf{q} = \mathbf{0}$ .

Second, we know that any solution to the system  $\mathbb{A}\mathbf{q} = \mathbf{0}$  can be written in the form  $(\mathbf{I} - \mathbb{A}^g \mathbb{A})\mathbf{p}$  [17]. As  $\mathbb{A}_2^g \mathbf{c} - \mathbb{A}_1^g \mathbf{c}$  is a solution to system  $\mathbb{A}\mathbf{q} = \mathbf{0}$ , there must exist a  $\mathbf{p}_0$  such that

$$(\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_0 = \mathbb{A}_2^g \mathbf{c} - \mathbb{A}_1^g \mathbf{c}. \quad (23)$$

In addition, it is easy to check that  $(\mathbf{I} - \mathbb{A}_1^g \mathbb{A})\mathbf{p}_1 + (\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_0$  is also a solution to the system  $\mathbb{A}\mathbf{q} = \mathbf{0}$ . Thus, there exists a  $\mathbf{p}_2$  such that

$$(\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_2 = (\mathbf{I} - \mathbb{A}_1^g \mathbb{A})\mathbf{p}_1 + (\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_0. \quad (24)$$

Plugging (23) into (24), we have

$$\begin{aligned} (\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_2 &= (\mathbf{I} - \mathbb{A}_1^g \mathbb{A})\mathbf{p}_1 + (\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_0 \\ &= (\mathbf{I} - \mathbb{A}_1^g \mathbb{A})\mathbf{p}_1 + \mathbb{A}_2^g \mathbf{c} - \mathbb{A}_1^g \mathbf{c} \end{aligned} \quad (25)$$

$$\preceq \mathbb{A}_2^g \mathbf{c}, \quad (26)$$

in which the last inequality comes from the assumption that  $(\mathbf{I} - \mathbb{A}_1^g \mathbb{A})\mathbf{p}_1 \preceq \mathbb{A}_1^g \mathbf{c}$ . Hence, we have found a  $\mathbf{p}_2$ , such that  $(\mathbf{I} - \mathbb{A}_2^g \mathbb{A})\mathbf{p}_2 \preceq \mathbb{A}_2^g \mathbf{c}$ . This implies that  $h_2^* = 0$ . ■

**Remark 1.** *The proposed algorithm for checking whether  $\text{Sim}_Y(Z \rightarrow X)$  holds or not has a polynomial complexity. Among all operations required, computing the  $g$ -inverse and solving the LP defined by (19) require most computations. The complexity to obtain  $\mathbb{A}^g$  is of order  $O(n^3)$  [18]. Furthermore, there exists polynomial complexity algorithms to solve the LP defined by (19). For example, [14] provided an algorithm to solve LP using  $O(n^3 L)$  operations, where  $L$  is number of binary bits needed to store input data of the problem (one can refer to Chapter 8 in [15] for more details about the complexity of algorithms for solving LP). Hence, the total operations of our algorithm for checking  $\text{Sim}_Y(Z \rightarrow X)$  is of order  $O(n^3 L)$ . In addition, we note that we can terminate the LP algorithm earlier once the algorithm finds a  $\mathbf{t}$  such that  $\mathbf{t} \mathbb{A}^g \mathbf{c} < 0$ , as this indicates that  $h^* < 0$ . This can potentially further reduce the computational complexity.*

Thus, we can conclude that the proposed algorithm can check whether  $\text{Sim}_Y(Z \rightarrow X)$  holds or not with a polynomial complexity. Algorithm 1 summarizes the main steps involved in our algorithm. In the following algorithm, we use  $\text{Res} = 0$  to denote that  $\text{Sim}_Y(Z \rightarrow X)$  does not hold and  $\text{Res} = 1$  to denote that  $\text{Sim}_Y(Z \rightarrow X)$  holds.

In the following, we provide our answer to the second open question, i.e., if  $\text{Sim}_Y(Z \rightarrow X)$  holds, how to find  $P_{\tilde{X}|Z}$  efficiently.

**Theorem 3.** *Let  $\mathbf{e}$  be any  $n \times 1$  vector with  $\mathbf{e} \succ \mathbf{0}$ , and  $\mathbf{q}^*$  be the obtained from the following LP:*

$$\begin{aligned} \min_{\mathbf{q}} \quad & f(\mathbf{q}) = \mathbf{e}^T \mathbf{q}, \\ \text{s.t.} \quad & \mathbf{q} \succeq \mathbf{0}, \\ & \mathbb{A} \mathbf{q} = \mathbf{c}. \end{aligned} \tag{27}$$

*If  $\text{Sim}_Y(Z \rightarrow X)$  holds, then  $\mathbf{Q}^* = \text{Reshape}(\mathbf{q}^*, [|\mathcal{X}|, |\mathcal{Z}|])^T$  is a valid choice for  $P_{\tilde{X}|Z}$ .*

*Proof:* By assumption,  $\text{Sim}_Y(Z \rightarrow X)$  holds, which implies that the system defined by (15)

---

**Algorithm 1** Checking  $\text{Sim}_Y(Z \rightarrow X)$ 

---

- 1: **Input:** PMF  $P_{XYZ}$ ;
  - 2: **Initiate:**
  - 3:   a. Calculate matrices  $\mathbf{A}$  and  $\mathbf{C}$ ;
  - 4:   b. Construct  $\mathbf{c}$  and  $\mathbb{A}$  using (10) and (11) respectively;
  - 5:   c. Set  $\text{Res} = 0$ ;
  - 6: **if** ( $\text{Rank}(\mathbb{A}) \neq \text{Rank}(\mathbb{A}|\mathbf{c})$ ) **then**
  - 7:   **break**;
  - 8: **else**
  - 9:   d. Find a  $\mathbb{A}^g$ , and calculate  $\mathbb{A}^g\mathbf{c}$ ,  $\mathbf{I} - \mathbb{A}^g\mathbb{A}$ ;
  - 10:   e. Solve LP (19) and obtain  $h^*$ ;
  - 11:   **if** ( $h^* == 0$ ) **then**
  - 12:      $\text{Res} = 1$ ;
  - 13:   **else**
  - 14:     **break**;
  - 15:   **end if**
  - 16: **end if**
  - 17: **Output:**  $\text{Res}$ .
- 

is consistent and it has nonnegative solutions. Hence, the following LP is feasible

$$\begin{aligned} \min_{\mathbf{q}} f(\mathbf{q}) &= \mathbf{e}^T \mathbf{q}, \\ \text{s.t.} \quad \mathbf{q} &\succeq \mathbf{0}, \\ \mathbb{A}\mathbf{q} &= \mathbf{c}, \end{aligned} \tag{28}$$

where  $\mathbf{e} \succ \mathbf{0}$ . Hence, the minimizer  $\mathbf{q}^*$  is nonnegative and satisfies  $\mathbb{A}\mathbf{q}^* = \mathbf{c}$ . We can then reshape  $\mathbf{q}^*$  into matrix  $\mathbf{Q}^*$  (see (12)).  $\mathbf{Q}^*$  is a valid choice for  $P_{\bar{X}|Z}$ . ■

**Remark 2.** Since finding a suitable  $P_{\bar{X}|Z}$  using our approach is equivalent to solving a LP, the

complexity is of polynomial order.

**Remark 3.** For a given distribution  $P_{XYZ}$ , there may be more than one possible  $P_{\bar{X}|Z}$  such that (1) holds. Different choices of  $\mathbf{e}$  in (27) give different values for  $P_{\bar{X}|Z}$ .

**Remark 4.** The objective function  $f(\mathbf{q})$  can be further modified to satisfy various design criteria of Eve. For example, let

$$\tilde{\mathbf{q}} = \text{Vec}(\tilde{\mathbf{Q}}[\tilde{q}_{kj}]^T)$$

with  $\tilde{q}_{kj} = P_{X|Z}(k|j)$ , then setting

$$f(\mathbf{q}) = \|\mathbf{q} - \tilde{\mathbf{q}}\|_2^2$$

will minimize the amount of changes in the conditional PMF in the  $l_2$  norm sense. This is a quadratic programming, which can still be solved efficiently.

#### IV. NUMERICAL EXAMPLES

In this section, we provide several examples to illustrate the proposed algorithm. We also use some of the examples used in [10] to compare our proposed algorithm with the method in [10].

**Example 1:** Let  $P_{XYZ}$  with ranges  $\mathcal{X} = \{x_1, x_2\}$ ,  $\mathcal{Y} = \{y_1, y_2\}$  and  $\mathcal{Z} = \{z_1, z_2, z_3\}$  be:

$$P_{XYZ}(x_1, y_1, z_1) = 6/100,$$

$$P_{XYZ}(x_2, y_1, z_1) = 4/100,$$

$$P_{XYZ}(x_1, y_1, z_2) = 9/100,$$

$$P_{XYZ}(x_2, y_1, z_2) = 6/100,$$

$$P_{XYZ}(x_1, y_1, z_3) = 15/100,$$

$$P_{XYZ}(x_2, y_1, z_3) = 10/100,$$

$$P_{XYZ}(x_1, y_2, z_1) = 36/100,$$

$$P_{XYZ}(x_2, y_2, z_1) = 4/100,$$

$$P_{XYZ}(x_1, y_2, z_2) = 9/100,$$

$$P_{XYZ}(x_2, y_2, z_2) = 1/100,$$

$$P_{XYZ}(x_1, y_2, z_3) = 0,$$

$$P_{XYZ}(x_2, y_2, z_3) = 0.$$

To use our algorithm, we have the following steps:

*Step 1:* Compute  $P_{YZ}$  and  $P_{YX}$ , and write them in the matrix form  $\mathbf{A}$  and  $\mathbf{C}$ :

$$\mathbf{A} = \begin{bmatrix} 0.1 & 0.15 & 0.25 \\ 0.4 & 0.1 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0.3 & 0.2 \\ 0.45 & 0.05 \end{bmatrix}. \quad (29)$$

*Step 2:* Construct  $\mathbb{A}$  and  $\mathbf{c}$  using (10) and (11) respectively:

$$\mathbb{A} = \begin{bmatrix} 0.1 & 0 & 0.15 & 0 & 0.25 & 0 \\ 0 & 0.1 & 0 & 0.15 & 0 & 0.25 \\ 0.4 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0.1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad (30)$$

$$\mathbf{c} = [0.3, 0.2, 0.45, 0.05, 1, 1, 1]^T. \quad (31)$$

*Step 3:* Check the ranks of  $\mathbb{A}$  and  $(\mathbb{A}|\mathbf{c})$ :

We get

$$\text{Rank}(\mathbb{A}) = \text{Rank}((\mathbb{A}|\mathbf{c})) = 5. \quad (32)$$

*Step 4:* Choose the g-inverse to be the Moore-Penrose pseudoinverse  $\mathbb{A}^+$  and calculate  $\mathbb{A}^+\mathbf{c}$  and

$\mathbf{I} - \mathbb{A}^+\mathbb{A}$ :

$$\mathbb{A}^+\mathbf{c} = \begin{bmatrix} 0.9762 \\ 0.0238 \\ 0.5952 \\ 0.4048 \\ 0.4524 \\ 0.5476 \end{bmatrix}, \quad (33)$$

$$\mathbf{I} - \mathbb{A}^+\mathbb{A} = \begin{bmatrix} 0.0238 & -0.0238 & -0.0952 & 0.0952 & 0.0476 & -0.0476 \\ -0.0238 & 0.0238 & 0.0952 & -0.0952 & -0.0476 & 0.0476 \\ -0.0952 & 0.0952 & 0.3810 & -0.3810 & -0.1905 & 0.1905 \\ 0.0952 & -0.0952 & -0.3810 & 0.3810 & 0.1905 & -0.1905 \\ 0.0476 & -0.0476 & -0.1905 & -0.1905 & 0.0952 & -0.0952 \\ -0.0476 & 0.0476 & 0.1905 & -0.1905 & -0.0952 & 0.0952 \end{bmatrix}. \quad (34)$$

*Step 5:* Solve LP (19). Using the above data, we obtain  $h^* = 0$ , which implies that  $\text{Sim}_Y(Z \rightarrow X)$  holds.

*Step 6:* Obtain a possible  $P_{\bar{X}|Z}$ . We construct the LP defined in (27) with  $\mathbf{e} = [2, 2, 2, 1, 1, 1]^T$ , and get

$$\mathbf{q}^* = [1, 0, 1/2, 1/2, 1/2, 1/2]^T.$$

Thus the simulatability channel is

$$P_{\bar{X}|Z} = \begin{bmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}, \quad (35)$$

which is consistent with the result obtained from the criterion proposed in [10]. If we set  $\mathbf{e} = [1, 1, 1, 1, 1, 1]^T$ , we get

$$\mathbf{q}^* = [0.9762, 0.0238, 0.5952, 0.4048, 0.4524, 0.5476]^T,$$

which implies that another valid choice is

$$P_{\bar{X}|Z} = \begin{bmatrix} 0.9762 & 0.0238 \\ 0.5962 & 0.4048 \\ 0.4524 & 0.5476 \end{bmatrix}. \quad (36)$$

**Example 2:** In this example, we consider a case in which  $Y$  is not binary. To represent the joint PMF concisely, we follow the same approach in [10] and use

$$M_{UV} = (P_U(u), (P_{V|U=u}(v_1), \dots, P_{V|U=u}(v_{|V|-1})))_{u \in \mathcal{U}}$$

to represent the joint PMF  $P_{UV}$ . For this example, we set

$$\begin{aligned} M_{ZY} &= (0.3, (0, 0)), (0.3, (0.5, 0)), \\ &\quad (0.3, (0.25, \sqrt{3}/4)), (0.1, (0.25, \sqrt{3}/12)), \\ M_{XY} &= (0.3, (0.25, 0)), (0.3, (0.375, \sqrt{3}/8)), \\ &\quad (0.3, (0.125, \sqrt{3}/8)), (0.05, (0.24, \sqrt{3}/12)), \\ &\quad (0.05, (0.26, \sqrt{3}/12)). \end{aligned} \quad (37)$$

In step 1, we write  $P_{YZ}$  and  $P_{YX}$  in the matrix form  $\mathbf{A}$  and  $\mathbf{C}$ :

$$\mathbf{A} = \begin{bmatrix} 0 & 0.1500 & 0.0750 & 0.0250 \\ 0 & 0 & 0.1299 & 0.0144 \\ 0.3000 & 0.1500 & 0.0951 & 0.0606 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 0.0750 & 0.1125 & 0.0375 & 0.0120 & 0.0130 \\ 0 & 0.0650 & 0.0650 & 0.0072 & 0.0072 \\ 0.2250 & 0.1225 & 0.1975 & 0.0308 & 0.0298 \end{bmatrix}.$$

To make the paper concise, we do not list the values of  $\mathbf{A}$ ,  $\mathbf{c}$  and following steps in details. Steps 2, 3, 4 are similar to those in Example 1. But in Step 5, we obtain that  $h^* < 0$ , which indicates that  $\text{Sim}_Y(Z \rightarrow X)$  does not hold. This result is also consistent with the conclusion in [10], which is obtained by an analysis that exploits the special mass constellation structure of the data. We note that the mechanical model based “more centered” criterion in [10] does not work for this example, as  $Y$  is not binary anymore, although the mass constellation representation of PMFs can still be used to exploit the special structure that this set of data has.

Next, we provide an example for which the mass constellation presentation does not work while our algorithm can easily obtain the answers.

**Example 3:** In this example, we consider  $X, Y, Z$  with larger dimensions, in particular, we set  $|\mathcal{X}| = 4$ ,  $|\mathcal{Y}| = 4$ , and  $|\mathcal{Z}| = 6$ . Again to represent the joint PMF concisely, we use the same method as that used in Example 2 to represent  $P_{XYZ}$ . For this example, we randomly set

$$M_{ZY} =$$

$$(0.1604, (0.1966, 0.1054, 0.4198)), (0.1654, (0.1230, 0.4709, 0.3355)),$$

$$(0.1613, (0.0350, 0.6219, 0.0823)), (0.1504, (0.4585, 0.2504, 0.2343)),$$

$$(0.1207, (0.2443, 0.4704, 0.0701)), (0.2419, (0.2979, 0.1151, 0.4601));$$

$$M_{XY} =$$

$$(0.2603, (0.1784, 0.3822, 0.2056)), (0.2181, (0.1538, 0.4409, 0.2255)),$$

$$(0.2356, (0.2129, 0.2684, 0.3913)), (0.2861, (0.3422, 0.2044, 0.3363)).$$



We denote the above PMF with following two matrices

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.0315 & 0.0203 & 0.0056 & 0.0690 & 0.0295 & 0.0720 \\ 0.0169 & 0.0779 & 0.1003 & 0.0377 & 0.0568 & 0.0278 \\ 0.0673 & 0.0555 & 0.0133 & 0.0352 & 0.0085 & 0.1113 \\ 0.0446 & 0.0117 & 0.0421 & 0.0085 & 0.0260 & 0.0307 \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} 0.0464 & 0.0335 & 0.0502 & 0.0979 \\ 0.0995 & 0.0962 & 0.0632 & 0.0585 \\ 0.0535 & 0.0492 & 0.0922 & 0.0962 \\ 0.0609 & 0.0392 & 0.0300 & 0.0335 \end{bmatrix}. \end{aligned} \quad (38)$$

Following the same steps as those in Example 1, we obtain that  $h^* = 0$ , which means  $\text{Sim}_Y(Z \rightarrow X)$  holds. Furthermore, by setting  $\mathbf{e} = \mathbf{1}_{24 \times 1}$  in (27), we obtain one possible  $P_{\bar{X}|Z}$ , denoted by matrix  $\mathbf{Q}^*$ :

$$\mathbf{Q}^* = \begin{bmatrix} 0.4979 & 0.1504 & 0.2038 & 0.1479 \\ 0.0148 & 0.3751 & 0.5618 & 0.0483 \\ 0.5210 & 0.4391 & 0.0254 & 0.0144 \\ 0.1302 & 0.0917 & 0.0301 & 0.7481 \\ 0.5638 & 0.2674 & 0.0161 & 0.1527 \\ 0.0261 & 0.0622 & 0.4110 & 0.5006 \end{bmatrix}. \quad (39)$$

One can easily check that  $\mathbf{A}\mathbf{Q}^* = \mathbf{C}$  holds. We note that, because of the lack of special data structure and the high dimensions, it is difficult to use the mass constellation structure of [10] to check whether  $\text{Sim}_Y(Z \rightarrow X)$  holds or not in this example.

**Example 4:** In this example, we consider the following PMF  $P_{XY}$ :

$$P_{XY}(x, y) = \begin{cases} \frac{1-\alpha}{2}, & \text{if } x = y; \\ \frac{\alpha}{2}, & \text{if } x \neq y, \end{cases}$$

and  $Z$  is generated by  $[X, Y]$  via an erasure channel with erasure probability  $1 - \gamma$ , i.e.,  $Z = (X, Y)$  with a probability  $\gamma$  and  $Z = \phi$  with probability  $1 - \gamma$ . It was shown in [10] that  $\text{sim}_Y(Z \rightarrow X)$  and  $\text{sim}_X(Z \rightarrow Y)$  hold if and only if  $\gamma \geq 1 - 2\alpha$ . In the following, we use our algorithm to verify the obtained result.

As above, in step 1, we compute  $P_{YZ}$  and write  $P_{YZ}$  and  $P_{YX}$  in matrix form  $\mathbf{A}$  and  $\mathbf{C}$ :

$$\mathbf{A} = \begin{bmatrix} \frac{(1-\alpha)\gamma}{2} & \frac{\alpha\gamma}{2} & 0 & 0 & \frac{1-\gamma}{2} \\ 0 & 0 & \frac{\alpha\gamma}{2} & \frac{(1-\alpha)\gamma}{2} & \frac{1-\gamma}{2} \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \frac{1-\alpha}{2} & \frac{\alpha}{2} \\ \frac{\alpha}{2} & \frac{1-\alpha}{2} \end{bmatrix}.$$

In step 2, we calculate matrices  $\mathbb{A}$  and  $\mathbf{c}$ :

$$\mathbb{A} = \begin{bmatrix} \frac{(1-\alpha)\gamma}{2} & 0 & \frac{\alpha\gamma}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1-\gamma}{2} & 0 \\ 0 & \frac{(1-\alpha)\gamma}{2} & 0 & \frac{\alpha\gamma}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1-\gamma}{2} \\ 0 & 0 & 0 & 0 & \frac{\alpha\gamma}{2} & 0 & \frac{(1-\alpha)\gamma}{2} & 0 & \frac{1-\gamma}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\alpha\gamma}{2} & 0 & \frac{(1-\alpha)\gamma}{2} & 0 & \frac{1-\gamma}{2} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$\mathbf{c} = [1 - \alpha, \alpha, \alpha, 1 - \alpha, 1, 1, 1, 1, 1]^T.$$

The following steps are similar to those in Examples 1 and 2. Using our algorithm, we can find that, for any given values  $\alpha$  and  $\gamma$ , as long as  $\gamma \geq 1 - 2\alpha$ ,  $h^* = 0$ , and the simulatability condition holds. We can also obtain a possible simulatability channel  $P_{\bar{X}|Z}$  that Eve may use, following the same steps as in Example 1. On the other side, if  $\gamma < 1 - 2\alpha$ , we obtained  $h^* < 0$ , and hence the simulatability condition does not hold.

## V. COMPLEXITY REDUCTION

In Proposition 1, we show that different choices of  $\mathbb{A}^g$  will not affect the result on whether  $h^*$  equals zero or not. However, different choices of  $\mathbb{A}^g$  may affect the amount of computation needed. Primal-dual path-following method is one of the best methods for solving LP of the

following form [15]:

$$\begin{aligned} & \min_{\mathbf{t}} \mathbf{t}^T \mathbf{b} \\ \text{s.t.} \quad & \mathbf{t} \succeq \mathbf{0}, \\ & \mathbf{B}\mathbf{t} = \mathbf{d}, \end{aligned}$$

in which  $\mathbf{B}$  is a matrix of size  $m \times n$ . The complexity is related to the size of  $\mathbf{B}$ . In particular, in terms of  $m$  and  $n$ , the complexity is  $O((nm^2 + n^{1.5}m)L)$  [19], [20]. In LP (19) constructed in the proof of Theorem 2,  $\mathbf{B} = (\mathbf{I} - \mathbb{A}^g \mathbb{A})^T$ , which is an  $n \times n$  matrix, and hence the complexity is  $O(n^3 L)$  as mentioned in Section III.

In the following, we show that if we choose the g-inverse of  $\mathbb{A}$  to be  $\mathbb{A}^+$ , the Moore-Penrose inverse, the problem size can be reduced by some further transformations. Let the SVD of  $\mathbb{A}$  be  $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ . Then  $\mathbb{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T$ . Suppose  $\text{rank}(\mathbf{\Sigma}_{m \times n}) = r$  and set  $s = n - r$ . We have

$$\begin{aligned} \mathbb{A}^+ \mathbb{A} &= \mathbf{V}\mathbf{\Sigma}^+ \mathbf{U}^T \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \\ &= \mathbf{V} \begin{bmatrix} \mathbf{I}_r & \mathbf{0}_{r \times s} \\ \mathbf{0}_{s \times r} & \mathbf{0}_{s \times s} \end{bmatrix} \mathbf{V}^T. \end{aligned} \quad (40)$$

As discussed in the proof of Theorem 2, checking  $\text{Sim}_Y(Z \rightarrow X)$  holds or not is equivalent to checking whether

$$(\mathbf{I} - \mathbb{A}^+ \mathbb{A})\mathbf{p} \preceq \mathbb{A}^+ \mathbf{c} \quad (41)$$

has a solution or not. We now perform some transformations on (41). First we have

$$\begin{aligned} \mathbf{I} - \mathbb{A}^+ \mathbb{A} &= \mathbf{V} \begin{bmatrix} \mathbf{I}_r & \mathbf{0}_{r \times s} \\ \mathbf{0}_{s \times r} & \mathbf{I}_s \end{bmatrix} \mathbf{V}^T - \mathbf{V} \begin{bmatrix} \mathbf{I}_r & \mathbf{0}_{r \times s} \\ \mathbf{0}_{s \times r} & \mathbf{0}_{s \times s} \end{bmatrix} \mathbf{V}^T \\ &= \mathbf{V} \begin{bmatrix} \mathbf{0}_{r \times r} & \mathbf{0}_{r \times s} \\ \mathbf{0}_{s \times r} & \mathbf{I}_s \end{bmatrix} \mathbf{V}^T. \end{aligned} \quad (42)$$

Hence, (41) is equivalent to

$$\mathbf{V} \begin{bmatrix} \mathbf{0}_{r \times r} & \mathbf{0}_{r \times s} \\ \mathbf{0}_{s \times r} & \mathbf{I}_s \end{bmatrix} \mathbf{V}^T \mathbf{p} \preceq \mathbb{A}^+ \mathbf{c}. \quad (43)$$

$\mathbf{V}$  can be split into four blocks as

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{r \times r} & \mathbf{V}_{r \times s} \\ \mathbf{V}_{s \times r} & \mathbf{V}_{s \times s} \end{bmatrix}. \quad (44)$$

We use  $\mathbf{w}$  to denote the  $n \times 1$  column vector  $\mathbf{V}^T \mathbf{p}$ , i.e.,

$$\mathbf{w} = \mathbf{V}^T \mathbf{p}. \quad (45)$$

Note that  $\mathbf{p} \leftrightarrow \mathbf{w}$  is a reversible bijection, since  $\mathbf{V}^T$  is a full rank matrix.

Then (43) is equivalent to

$$\begin{bmatrix} \mathbf{0}_{r \times r} & \mathbf{V}_{r \times s} \\ \mathbf{0}_{s \times r} & \mathbf{V}_{s \times s} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{r \times 1} \\ \mathbf{w}_{s \times 1} \end{bmatrix} \preceq \mathbb{A}^+ \mathbf{c}, \quad (46)$$

which is equivalent to

$$\begin{bmatrix} \mathbf{V}_{r \times s} \\ \mathbf{V}_{s \times s} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{s \times 1} \end{bmatrix} \preceq \mathbb{A}^+ \mathbf{c}. \quad (47)$$

Hence, checking whether (41) has a solution or not is equivalent to checking whether (47) has a solution or not. To check whether (47) has a solution or not, we can construct a new LP for (47) in the same way as in the proof in Theorem 2. However, the size of the newly constructed LP will be smaller than that of (19) constructed in the proof of Theorem 2. The complexity for the newly constructed LP will be  $O((ns^2 + n^{1.5}s)L)$ . Since  $s$  is always less than or equal to  $n$  (sometimes,  $s$  can be much less than  $n$ ) and that  $L$  doesn't change, compared with the LP (19), the computational complexity for this new LP will be reduced.

## VI. CONCLUSION

In this paper, we have proposed an efficient algorithm to check the simulatability condition, an important condition in the problems of secret key generation using a non-authenticated public channel. We have also proposed a simple and flexible method to calculate a possible simulatability channel if the simulatability condition holds. The proposed algorithms have polynomial complexities. We have presented numerical examples to show the efficiency of the protocol. Finally, we have proposed an approach to further reduce the computational complexity.

## APPENDIX A

### FARKAS' LEMMA

There are several equivalent forms of the Farkas' lemma [12]. Here, we state a form that will be used in our proof.

**Lemma 1.** (*Farkas' Lemma [12]*) *Let  $\mathbf{B}$  be a matrix, and  $\mathbf{b}$  be a vector, then the system specified by  $\mathbf{B}\mathbf{p} \preceq \mathbf{b}$ , has a solution  $\mathbf{p}$ , if and only if  $\mathbf{t}^T \mathbf{b} \geq 0$  for each column vector  $\mathbf{t} \succeq \mathbf{0}$  with  $\mathbf{B}^T \mathbf{t} = \mathbf{0}$ .*

## REFERENCES

- [1] U. Maurer, “Secret key agreement by public discussion from common information,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 733–742, May 1993.
- [2] R. Ahlswede and I. Csiszár, “Common randomness in information theory and cryptography, Part I: Secret sharing,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 1121–1132, July 1993.
- [3] I. Csiszár and P. Narayan, “Common randomness and secret key generation with a helper,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 344–366, Mar. 2000.
- [4] S. Nitinawarat, C. Ye, A. Barg, P. Narayan, and A. Reznik, “Secret key generation for a pairwise independent network model,” *IEEE Trans. Inform. Theory*, vol. 56, pp. 6482–6489, Dec. 2010.
- [5] C. Chan and L. Zheng, “Network coding for secret key agreement,” *IEEE Trans. Inform. Theory*, 2010. Submitted.
- [6] I. Csiszár and P. Narayan, “Secrecy capacities for multiple terminals,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 3047–3061, Dec. 2004.
- [7] C. Ye and P. Narayan, “Secret key and private key constructions for simple multiterminal source models,” *IEEE Trans. Inform. Theory*, vol. 58, pp. 639–651, Feb. 2012.
- [8] U. Maurer, “Information-theoretically secure secret-key agreement by not authenticated public discussion,” in *Advances in CryptologyEurocrypt97*, pp. 209–225, Springer, 1997.
- [9] U. M. Maurer and S. Wolf, “Secret key agreement over a non-authenticated channel - Part I: Definitions and bounds,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 822–831, Apr. 2003.
- [10] U. M. Maurer and S. Wolf, “Secret key agreement over a non-authenticated channel - Part II: The simulatability condition,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 832–838, Apr. 2003.
- [11] U. M. Maurer and S. Wolf, “Secret key agreement over a non-authenticated channel - Part III: Privacy amplification,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 839–851, Apr. 2003.
- [12] A. Schrijver, *Theory of linear and integer programming*. New York: John Wiley & Sons, 1998.
- [13] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311, ACM, 1984.
- [14] C. C. Gonzaga, *An algorithm for solving linear programming problems in  $O(n^3 L)$  operations*. New York: Springer, 1989.
- [15] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*. New York: John Wiley & Sons, 2011.
- [16] C. R. Rao and S. K. Mitra, *Generalized inverse of matrices and its applications*. New York: John Wiley & Sons, 1971.
- [17] C. R. Rao, “Calculus of generalized inverses of matrices Part I: General theory,” *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 317–342, 1967.
- [18] H.-M. Möller, *Exact Computation of the Generalized Inverse and the Least-squares Solution*. Techn. Univ., Fak. für Mathematik, 1999.
- [19] R. D. Monteiro and I. Adler, “Interior path following primal-dual algorithms. Part I: Linear programming,” *Mathematical Programming*, vol. 44, pp. 27–41, 1989.
- [20] R. D. Monteiro and I. Adler, “Interior path following primal-dual algorithms. Part II: Convex quadratic programming,” *Mathematical Programming*, vol. 44, pp. 43–66, 1989.